

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (previously presented) A microprocessor for processing various assembler codes, comprising:

a parameter designating a respective assembler code and, depending on how the parameter is set, a different relative addressing takes place; and

a plurality of program counters and, dependent on the parameter, for each of said respective assembler codes one of said program counters is active in a computation of relative addresses.

2. (original) The microprocessor according to claim 1, including:

a computation unit; and

a multiplexer connected to said program counters, said multiplexer receives and is controlled by the parameter, said multiplexer having an output connected to said computation unit for the relative addresses.

Applic. No. 09/928,011
Response Dated May 13, 2005
Responsive to Office Action of January 13, 2005

3. (original) A microprocessor for processing various assembler codes, comprising:

a multiplexer having a first input, a second input for receiving a 0 value, and a third input receiving a parameter designating a respective assembler code and, depending on how the parameter is set, a different relative addressing takes place;

a program counter;

a computation unit for computing relative addresses;

an adding unit connected between said program counter and said computation unit, said adding unit having a first input connected to said program counter, a second input connected to said multiplexer, and an output connected to said computation unit; and

a memory for storing an instruction length and having an output connected to said first input of said multiplexer.

4. (original) A microprocessor for processing various assembler codes, comprising:

Applic. No. 09/928,011
Response Dated May 13, 2005
Responsive to Office Action of January 13, 2005

a multiplexer having a first input, a second input for receiving a 0 value, and a third input receiving a parameter designating a respective assembler code and, depending on how the parameter is set, a different relative addressing takes place;

a program counter;

a computation unit for computing relative addresses;

an subtracting unit connected between said program counter and said computation unit for the relative addresses, said subtracting unit having a first input connected to said program counter, a second input connected to said multiplexer, and an output connected to said computation unit; and

a memory for storing an instruction length and having an output connected to said first input of said multiplexer.

5. (original) A method of relative addressing in a microprocessor, which comprises the steps of:

Applic. No. 09/928,011
Response Dated May 13, 2005
Responsive to Office Action of January 13, 2005

determining relative addresses in dependence on one of an operating state and a parameter for a respective assembler code;

providing a plurality of program counters for various operating states and assembler codes; and

selecting one of the program counters for use in determining the relative addresses in dependence on one of the operating state and the respective assembler code.

6. (currently amended) The method according to claim 5, which comprises performing one of:

performing one of an addition and a subtraction of an instruction length to/from [[a]] the selected one of the program ~~counter~~-counters' reading for a relative address computation in dependence on one of the various operating states and the assembler codes; and

leaving the selected one of the program ~~counter~~-counters' reading unchanged.

7. (original) The method according to claim 6, which comprises performing one of:

Applic. No. 09/928,011
Response Dated May 13, 2005
Responsive to Office Action of January 13, 2005

performing one of an addition and a subtraction of the
instruction length to/from an offset value used for the
computation of the relative addresses in dependence on one of
the various operating states and the assembler codes; and

leaving the offset value unchanged.